UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/611,291 | 06/30/2003 | R. Hugo Patterson | 6368P002 | 2915 |

8791 7590 11/01/2007

BLAKELY SOKOLOFF TAYLOR & ZAFMAN
1279 OAKMEAD PARKWAY
SUNNYVALE, CA 94085-4040

| EXAMINER | |
|---|---|
| STACE, BRENT S | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2161 | |

| MAIL DATE | DELIVERY MODE |
|---|---|
| 11/01/2007 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 10/611,291 | PATTERSON, R. HUGO | |
| | Examiner | Art Unit | |
| | Brent S. Stace | 2161 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *16 August 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1,3-11,13-18,20-22,24-33,35-43 and 45-50* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-11,13-18,20-22,24-33,35-43 and 45-50* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *11 May 2006* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application.
6) ☐ Other: _____.

## DETAILED ACTION

### *Remarks*

1. This communication is responsive to the amendment filed August 16th, 2007. Claims 1, 3-11, 13-18, 20-22, 24-33, 35-43, and 45-50 are pending. In the amendment filed August 16th, 2007, Claims 1, 8, 15, 20, 27, 31, 33, 40, and 47 are amended, and Claims 1, 8, 15, 20, 27, 31, 33, 40, and 47 are independent. The examiner acknowledges that no new matter was introduced and the claims are supported by the specification. This action is made FINAL.

### *Response to Arguments*

2. The Applicant's arguments filed August 16th, 2007 with respect to Claims 1, 3-11, 13-18, 20-22, 24-33, 35-43, and 45-50 have been considered but are not persuasive.

3. The examiner would like to note that Zwilling shrinking files before applying Hitz to take (any iteration of) a snapshot of the shrunken files is the correct paradigm for considering the Zwilling and Hitz references in combination.

4. As to the applicant's arguments with respect to Claims 1, 8, 15, 20, 27, 31, 33, 40, and 47 for the prior art(s) allegedly not teaching "[performing] garbage collecting in a storage device," the examiner respectfully disagrees. Zwilling does garbage collection because used parts of a file are moved to unused allocated parts of the file and any unreferenced parts of a file are deallocated. This enables a smaller size for a file and can be seen as removing the garbage (unreferenced and allocated units) from a file. As

shown below, mostly in Zwilling, col. 5, lines 29-52, Zwilling locates used/referenced data units, moves/copies them to an unallocated area, then unallocates the units that were used as the source for copying/moving. Applicant's rebuttal dated August 16th, 2007 allege that Zwilling still does not teach garbage collection essentially since Zwilling's allocation units are empty. The applicant has crafted a definition for garbage in that garbage refers to any object or data within a program's memory space that is unreachable from the programs root set and cannot ever be accessed by a running program. Zwilling still teaches garbage collection based on applican'ts crafted definition. Allocated and unused allocation units in Zwilling (empty or not (we will consider them empty for sake of this argument)) still contain at least an object or data. First, the unused allocation unit alone can be considered an object. Second, even if the unused allocation unit is empty, it will still hold data (even if it is all 0's). Internally, a computer uses binary to represent everything. Therefore, an unused allocated allocation unit, even if empty will contain either 0's, 1's or combination of 0's and 1's (since these are the only numbers a computer can use). Storing a void vacuum can not be done on a computer. Also, applicant's definition alone is erroneous in that it states that the object or data cannot ever be accessed by a running program. The object or data MUST be accessed at one time during garbage collection in order to determine to delete the unit or accessing it to delete it. Lastly, the applicant's definition appears to be directed towards garbage collection for a running program (like in JAVA) which does not appear to be the appropriate definition for the application since the application is directed toward garbage collection for file allocation/data management (for example, p.

22 of applicant's remarks file August 16th, 2007). Since the Applicant's definition is flawed and Zwilling still teaches garbage collection based on this definition, Zwilling is still seen as teaching garbage collection.

In responding to the Applicant's analogy set forth on p. 21 of the remarks dated August 16th, 2007, according to the examiner's interpretation of the art, the analogy would be placing empty garbage bags (allocated, unused allocation units) at the end of one's driveway to be picked up by the garbage collectors.

5. As to the applicant's arguments with respect to Claims 1, 8, 15, 20, 27, 31, 33, 40, and 47 for the prior art(s) allegedly not teaching "whether an allocation unit is unreferenced/inactive or not," the examiner respectfully disagrees. Zwilling must have a determination of a block being used/active/referenced and unused/inactive/unreferenced in order to copy the allocation units and guarantee file integrity. This is also shown in Zwilling, col. 12, line 37 "not in use" and Zwilling, col. 5, lines 37 "if an allocation unit...is used."

In response to the Applicant's remarks on p. 21 dated August 16th, 2007 regarding an alleged mis-correspondence between "used" with the concept of "active/referenced" blocks of data (allocation units) and "unused" with the concept of "unreferenced/inactive." Applicant's own specification at paragraph [0003] (last sentence) appears to equate these in stating "Therefore, garbage collection operations are performed within these backup systems to delete data that is no longer active/referenced, thereby reclaiming unused space in the backup storage device." "Data that is no longer active" is inactive data and "data that is no longer...referenced" is

unreferenced data. Therefore, the citing can be re-written as "Therefore, garbage collection operations are performed within these backup systems to delete data that is inactive/unreferenced, thereby reclaiming unused space in the backup storage device." Since Applicant's own specification in the background (prior art) section equates inactive/unreferenced with unused (as done by the examiner), the opposite can also be equated (used/active/referenced). As such, the Applicant's own specification appears to contradict and overcome the applicant's arguments.

6. As to the applicant's arguments with respect to Claims 1, 20, 27, 31, and 33 for the claims allegedly requiring "the opposite order of operations of performing garbage collection on snapshots," (subject matter the Applicant alleges the prior art(s) applied with the paradigm used for the combination (above) does not teach) the examiner respectfully disagrees. The examiner does not see subject matter in the claims requiring steps in a certain order. Also, considering Hitz does multiple snapshots, each garbage collected prior to a snapshot (as combined). The sequence of garbage collect, snapshot, ...garbage collect, snapshot, and so forth shows 2 orders of operations (garbage collect, snapshot and ...snapshot, garbage collect (the opposite of the first)). As such the reference appear to teach applicant's argument.

7. As to the applicant's arguments with respect to Claims 1, 8, 15, 20, 27, 31, 33, 40, and 47 for the prior art(s) allegedly not disclosing "blocks of data that are both referenced by at least one other block of data residing within a set of one or more storage trees," the examiner respectfully disagrees. Zwilling, col. 12, lines 24-53 with Zwilling, Fig. 2 with Zwilling, col. 8, lines 49-51 with Zwilling, col. 8, lines 60-64 with

Zwilling, Figs. 3C and 3E was used to reject this limitation. Regarding the "block(s) of data residing within a set of one or more storage trees" limitation, the Applicant at least admitted to Zwilling teaching at least a single binary tree (applicant's remarks of 2/21/07, p. 35, approx. middle of page). The binary tree in Zwilling has pages containing the data records that form leaf nodes of the tree (Zwilling, col. 8, lines 49-51 as noted from applicant's remarks of 2/21/07). At this point in this response we have a binary tree with nodes (so at least 2 nodes total in a tree). A tree with one node (root node) is still a tree in a tree structure. This is the most basic form of a tree. Any nodes descending from the root node can be considered as sub-trees within the main tree. These sub-trees are still trees. Therefore, the nodes of the binary tree in Zwilling, make trees since all the nodes of a tree can be considered trees by themselves. They are considered storage trees since they store information about the files and of the files. Depending on what subtrees within the main tree can be considered as trees by alone (there are at least as many trees as there are nodes), you are left with trees containing a plurality of nodes (especially considering Zwilling is implying the use a many nodes in the tree since there are many allocation units (Zwilling, col. 12, lines 24-53)). Alternatively, Zwilling has this tree for each file, and since Zwilling is doing operation on many files, this makes "a set of one or more storage trees." Regarding the "blocks of data that are both referenced by at least one other block of data," as shown above, Zwilling has multiple trees, and the trees doubly-linked nature makes blocks of data that are both referenced by at least one other block of data (especially considering that log records are sequentially ordered data records). Other interpretations of how the prior

art(s) maps to the claim may exist (as such the entirety of the references should be reviewed).

8. The other claims argued merely because of a dependency on a previously argued claim(s) in the arguments presented to the examiner, filed August 16th, 2007, are moot in view of the examiner's interpretation of the claims and art and are still considered rejected based on their respective rejections from a previous Office action (part(s) of recited again below).

### *Response to Amendment*

### *Specification*

9. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

### *Claim Objections*

10. In light of the applicant's respective arguments or respective amendments, some previous claim objections to the claims have been withdrawn.

### *Claim Rejections - 35 USC § 101*

11. In light of the applicant's respective arguments or respective amendments, some previous 35 USC § 101 rejections to the claims have been withdrawn.

***Claim Rejections - 35 USC § 103***

12. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

13. Claims 1, 3-7, 18, 20-22, 24-33, 35-39, and 50 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,249,792 (Zwilling et al.) in view of U.S. Patent No. 5,963,962 (Hitz et al.).

For **Claim 1**, Zwilling teaches: "A method of garbage collecting in a storage device [Zwilling, col. 5, lines 29-52 with Zwilling, Fig. 2] comprising:

- locating blocks of data in a log that are both referenced by at least one other block of data residing within a set of one or more storage trees, and within a range at a tail of the log using pruned walking, the range representing an address range within an allocated segment of the log, [Zwilling, col. 12, lines 24-53 with Zwilling, Fig. 2 with Zwilling, col. 8, lines 49-51 with Zwilling, col. 8, lines 60-64 with Zwilling, Figs. 3C and 3E]
- wherein the log is implemented in a hierarchical architecture [Zwilling, cols. 8-9, lines 46-26 with Zwilling, col. 12, lines 31-34]

- ...copying the blocks of data that are referenced...and within the range to an unallocated segment of the log [Zwilling, col. 5, lines 34-52 with Zwilling, col. 12, lines 41-53]
- ...marking the range at the tail of the log as unallocated so that at least a portion of an address space within the range can be reclaimed" [Zwilling, col. 5, lines 34-52 with Zwilling, col. 12, lines 31-34].

Zwilling discloses the above limitations but does not expressly teach:

- "...having a plurality of storage trees, each storage tree representing a snapshot taken at a point in time of target data being processed, each storage tree having a plurality of nodes and each node representing a block of data of the snapshot associated with each storage tree;
- ...by one or more other blocks of data of other nodes...wherein blocks of data that are not referenced by other blocks of data and within the range remain untouched."

With respect to Claim 1, an analogous art, Hitz, teaches:

- "...having a plurality of storage trees, [Hitz, cols. 17-18, lines 66-16] each representing a snapshot taken at a point in time of target data being processed, [Hitz, col. 17, lines 40-49] each storage tree having a plurality nodes [Hitz, cols. 17-18, lines 66-16] and each node representing a segment of data of the snapshot associated with each storage tree; [Hitz, col. 17, lines 40-49 with Hitz, cols. 17-18, lines 66-16]

- ...by one or more other blocks of data of other nodes [Hitz, col. 18, lines 35-38]

  ...wherein blocks of data that are not referenced by other blocks of data and within the range remain untouched" [Hitz, col. 15, lines 55-57 with Hitz, col. 16, lines 15-17 with Hitz, col. 20, lines 25-35 with Zwilling, col. 5, lines 29-52 with Zwilling, col. 6, lines 42-49 with Zwilling, col. 12, lines 36-37].

It would have been obvious to one of ordinary skill in the art at the time of invention having the teachings of Hitz and Zwilling before him/her to combine Hitz with Zwilling because both inventions are directed towards conserving file space (Zwilling shrinks and Hitz only snapshots data that was changed at a block level (not a file level)).

Hitz's invention would have been expected to successfully work well with Zwilling's invention because both inventions use file systems on computers. Zwilling discloses an on-line dynamic file shrink facility comprising shrinking log files. However, Zwilling does not expressly disclose storage trees each representing a snapshot and where unreferenced data remains untouched. Hitz discloses a write anywhere file-system layout comprising storage trees (file system snapshot trees) each representing a snapshot and where unreferenced data remains untouched.

It would have been obvious to one of ordinary skill in the art at the time of invention having the teachings of Hitz and Zwilling before him/her to take the write anywhere file-system layout from Hitz and install it into the invention of Zwilling, thereby offering the obvious advantage of taking snapshots of garbage collected/shrunken data, thus saving space. Shrinking the files prior to snapshoting them frees more space in Hitz thus creating more room in Hitz so that another later snapshot won't be prematurely

forced out when space in Hitz runs out (by snapshots consuming unacceptable numbers of disk blocks). Also, shinking prior to snapshoting guarantees that Hitz will not back up empty blocks (since they are dirtied by becoming empty in a snapshot). This also conserves space in Hitz. In fact, Hitz desires files to be in a shrunken form (Hitz, col. 5, lines 37-40). Therefore, Hitz is merely using Zwilling's shrinking technique to obtain files that have no fragments.

Different embodiments of Zwilling are use in the rejection for Claim 1 and its respective dependant claims. Zwilling (as modified by Hitz) teaches all of Claim 1 with assistance from Zwilling's different embodiments.

**Claim 3** can be mapped to Zwilling (as modified by Hitz) as follows: "The method of claim 1, wherein locating the blocks of data that are referenced and within the range includes determining a minimum value among addresses of descendent nodes of a node, [Zwilling, cols. 8-9, lines 46-14] wherein the minimum value represents a minimum address offset of a node that is closest referenced from the blocks of data" [Zwilling, cols. 8-9, lines 46-14 with Zwilling, col. 5, lines 29-52 with Zwilling Fig. 2].

**Claim 4** can be mapped to Zwilling (as modified by Hitz) as follows: "The method of claim 3, wherein a location table includes an entry for nodes that reference other nodes [Zwilling, col. 8, lines 46-64 with Zwilling, col. 10, lines 5-13] and wherein determining the minimum value among addresses of descendent nodes of the node includes retrieving the minimum value from an entry in the location table associated with the node" [Zwilling, cols. 8-9, lines 46-14 with Zwilling, col. 10, lines 5-13].

**Claim 5** can be mapped to Zwilling (as modified by Hitz) as follows: "The method of claim 4, wherein locating the blocks of data that are referenced and within the range includes processing the descendent nodes of the node upon determining that the minimum value among the addresses of the descent nodes is within the range" [Zwilling, col. 5, lines 29-52].

**Claim 6** can be mapped to Zwilling (as modified by Hitz) as follows: "The method of claim 5 comprising modifying the addresses of the copied blocks of data that are stored in the location table based on the new locations of the copied blocks of data in the log" [Zwilling, Fig. 3C with Zwilling, col. 8, lines 29-33 with Zwilling, col. 9, lines 4-7].

**Claim 7** can be mapped to Zwilling (as modified by Hitz) as follows: "The method of claim 5 further comprising modifying the minimum value in the entry in the table associated with the node when the minimum value changes based on the new locations of the copied blocks of data that are associated with descendent nodes of the node" [Zwilling, Fig. 3C with Zwilling, col. 8, lines 29-33 with Zwilling, col. 9, lines 4-7 with Zwilling, col. 5, lines 29-52 with Zwilling, col. 10, lines 5-13].

For **Claim 18**, Zwilling teaches: "The method of claim 15."

Zwilling discloses the above limitation but does not expressly teach: "...wherein at least one block of data stored in the log is referenced by more than one of other blocks of data."

With respect to Claim 18, an analogous art, Hitz, teaches: "...wherein at least one block of data stored in the log is referenced by more than one of other blocks of data" [Hitz, col. 18, lines 24-30 with Hitz, col. 2, lines 25-36].

It would have been obvious to one of ordinary skill in the art at the time of invention to combine Hitz with Zwilling because both inventions are directed towards storing files on file systems.

Hitz's invention would have been expected to successfully work well with Zwilling's invention because both inventions use file systems on computers. Zwilling discloses an on-line dynamic file shrink facility comprising trees/tables for file information/representation, however Zwilling does not expressly disclose the possibility that the blocks of data could be repeating by what is known in the art as aliases, shortcuts, or symbolic links. Hitz discloses a write anywhere file-system layout comprising file block indirection.

It would have been obvious to one of ordinary skill in the art at the time of invention to take the file block indirection from Hitz and install it into the method of Zwilling, thereby offering the obvious advantage of extending Zwilling's invention to work on files that contain the same data in attempts to save space as the snapshot size increases thereby increasing the number of active snapshots in Hitz.

For **Claim 20**, Zwilling teaches: "A system comprising:

- a storage device to store a number of blocks of data, [Zwilling, col. 5, lines 6-12] wherein the blocks of data that are marked as allocated are non-modifiable, [Zwilling, col. 6, lines 5-10] the blocks of data to be stored as a log; [Zwilling, col. 12, lines 15-17] and
- a garbage collection logic to locate the blocks of data that are both referenced by at least one other block within a set of one or more storage trees, and within a

range at a tail of the log using pruned walking, the range representing and address range within an allocated segment of the log, [Zwilling, col. 12, lines 24-53 with Zwilling, Fig. 2 with Zwilling, col. 8, lines 49-51 with Zwilling, col. 8, lines 60-64 with Zwilling, Figs. 3C and 3E]

- wherein the log is implemented in a hierarchical architecture [Zwilling, cols. 8-9, lines 46-26 with Zwilling, col. 12, lines 31-34]
- and wherein said garbage collection logic is operable to copy the blocks of data that are referenced and within the range at the fail of the log to a head of the log and mark the range as unallocated so that at least a portion of the address space within the range can be reclaimed" [Zwilling, col. 5, lines 29-52 with Zwilling, Fig. 2].

Zwilling discloses the above limitations but does not expressly teach:

- "...having a plurality of storage trees, each representing a snapshot taken at a point in time of target data being processed, each storage tree having a plurality nodes, and each node representing a block of data of the snapshot associated with each storage tree."

With respect to Claim 20, an analogous art, Hitz, teaches:

- "...having a plurality of storage trees, [Hitz, cols. 17-18, lines 66-16] each representing a snapshot taken at a point in time of target data being processed, [Hitz, col. 17, lines 40-49] each storage tree having a plurality nodes, [Hitz, cols. 17-18, lines 66-16] and each node representing a block of data of the snapshot

associated with each storage tree" [Hitz, col. 17, lines 40-49 with Hitz, cols. 17-18, lines 66-16].

It would have been obvious to one of ordinary skill in the art at the time of invention having the teachings of Hitz and Zwilling before him/her to combine Hitz with Zwilling because both inventions are directed towards conserving file space (Zwilling shrinks and Hitz only snapshots data that was changed at a block level (not a file level)).

Hitz's invention would have been expected to successfully work well with Zwilling's invention because both inventions use file systems on computers. Zwilling discloses an on-line dynamic file shrink facility comprising shrinking log files. However, Zwilling does not expressly disclose storage trees each representing a snapshot. Hitz discloses a write anywhere file-system layout comprising storage trees (file system snapshot trees) each representing a snapshot.

It would have been obvious to one of ordinary skill in the art at the time of invention having the teachings of Hitz and Zwilling before him/her to take the write anywhere file-system layout from Hitz and install it into the invention of Zwilling, thereby offering the obvious advantage of taking snapshots of garbage collected/shrunken data, thus saving space. Shrinking the files prior to snapshoting them frees more space in Hitz thus creating more room in Hitz so that another later snapshot won't be prematurely forced out when space in Hitz runs out (by snapshots consuming unacceptable numbers of disk blocks). Also, shinking prior to snapshoting guarantees that Hitz will not back up empty blocks (since they are dirtied by becoming empty in a snapshot). This also conserves space in Hitz. In fact, Hitz desires files to be in a shrunken form

(Hitz, col. 5, lines 37-40). Therefore, Hitz is merely using Zwilling's shrinking technique to obtain files that have no fragments.

Different embodiments of Zwilling are use in the rejection for Claim 20 and its respective dependant claims. Zwilling (as modified by Hitz) teaches all of Claim 20 with assistance from Zwilling's different embodiments.

**Claim 21** can be mapped to Zwilling (as modified by Hitz) as follows: "The system of claim 20, wherein the garbage collection logic is to copy the blocks of data that are referenced to an unallocated address space of the log" [Zwilling, col. 5, lines 29-52 with Zwilling, Fig. 2].

**Claim 22** can be mapped to Zwilling (as modified by Hitz) as follows: "The system of claim 21, wherein the garbage collection logic is to copy the blocks of data that are referenced to a head of the log" [Zwilling, col. 5, lines 29-52 with Zwilling, Fig. 2].

**Claim 24** can be mapped to Zwilling (as modified by Hitz) as follows: "The system of claim 20 wherein at least one of the number of blocks of data are referenced by more than one reference" [Hitz, col. 18, lines 24-30 with Hitz, col. 2, lines 25-36 with Hitz, col. 20, lines 25-35].

**Claim 25** can be mapped to Zwilling (as modified by Hitz) as follows: "The system of claim 20 comprising a location table to include entries associated with interior nodes of a storage tree, [Zwilling, col. 10, lines 5-13] wherein each entry is to include a minimum value among the addresses of descendent nodes of the associated interior

node" [Zwilling, cols. 8-9, lines 46-15 with Zwilling, col. 9, lines 43-51 with Zwilling col. 8, lines 20-35 with Zwilling, col. 7, lines 38-21].

**Claim 26** can be mapped to Zwilling (as modified by Hitz) (as modified by ) as follows: "The system of claim 25, wherein the garbage collection logic is to locate the blocks of data that are referenced and within the range at the tail of the log based on the minimum values stored in the entries of the location table" [Zwilling, col. 5, lines 29-40 with Zwilling, col. 8, lines 46-65 with Zwilling, col. 10, lines 5-13].

For **Claim 27**, Zwilling teaches:

- "...each leaf node of said plurality of storage trees to include a block of data from said file system [Zwilling, col. 10, lines 5-13 with Zwilling, col. 8, lines 46-65]
- a storage space to store said blocks of data having been allocated in said set of one or more storage devices; [Zwilling, col. 5, lines 29-52 with Zwilling, Fig. 2]
- having stored therein a minimum address value of descendent nodes of interior nodes of said plurality of storage trees; [Zwilling, cols. 8-9, lines 46-15 with Zwilling, col. 9, lines 43-51 with Zwilling col. 8, lines 20-35 with Zwilling, col. 7, lines 38-21] and
- a garbage collection logic to clean a currently selected range from the tail of said log, [Zwilling, col. 5, lines 29-52 with Zwilling, Fig. 2] said garbage collection logic to perform pruned walking of nodes of said plurality of storage trees based on said set of location tables and said currently selected range to locate blocks of data that are referenced by at least one other block of data currently residing within the plurality of storage trees" [Zwilling, cols. 8-9, lines 46-17 with Zwilling,

col. 5, lines 29-52 with Zwilling, col. 12, lines 24-53 with Zwilling, Fig. 2 with Zwilling, Figs. 3C and 3E].

Zwilling discloses the above limitations but does not expressly teach: "A backup system comprising:

- a plurality of storage trees, each representing a snapshot of a file system at a different time, each storage tree having a plurality of leaf nodes, ...that has been backed up from a set of one or more storage devices
- from a backup storage space
- a set of one or more location tables."

With respect to Claim 27, an analogous art, Hitz, teaches: "A backup system [Hitz, col. 17, lines 40-50] comprising:

- a plurality of storage trees, each representing a snapshot of a file system at a different time, each storage tree having a plurality of leaf nodes, [Hitz, cols. 17-18, lines 50-14 with Zwilling, col. 10, lines 49-56]...that has been backed up from a set of one or more storage devices [Hitz, cols. 17-18, lines 65-14]
- from a backup storage space [Hitz, cols. 17-18, lines 65-14]
- a set of one or more location tables" [Hitz, cols. 17-18, lines 65-14 with Zwilling, col. 10, lines 5-13].

It would have been obvious to one of ordinary skill in the art at the time of invention having the teachings of Hitz and Zwilling before him/her to combine Hitz with Zwilling because both inventions are directed towards conserving file space (Zwilling shrinks and Hitz only snapshots data that was changed at a block level (not a file level)).

Hitz's invention would have been expected to successfully work well with Zwilling's invention because both inventions use file systems on computers. Zwilling discloses an on-line dynamic file shrink facility comprising a storage tree, location table, and garbage collection, however Zwilling does not expressly disclose storage trees, or tables as relating to snapshots of storage device(s). Hitz discloses a write anywhere file-system layout comprising storage trees from snapshots of a file system.

It would have been obvious to one of ordinary skill in the art at the time of invention to take the storage device (making 2 storage devices), the storage trees of snapshots, and tables from Hitz and install it into the system of Zwilling, thereby offering the obvious advantage of extending Zwilling's invention to work on archived (snapshot) files in attempts to save space as the snapshot size increases thereby increasing the potential number of active snapshots in Hitz. Shrinking the files prior to snapshoting them frees more space in Hitz thus creating more room in Hitz so that another later snapshot won't be prematurely forced out when space in Hitz runs out (by snapshots consuming unacceptable numbers of disk blocks). Also, shinking prior to snapshoting guarantees that Hitz will not back up empty blocks (since they are dirtied by becoming empty in a snapshot). This also conserves space in Hitz. In fact, Hitz desires files to be in a shrunken form (Hitz, col. 5, lines 37-40). Therefore, Hitz can be seen as merely using Zwilling's shrinking technique to obtain files that have no fragments.

Different embodiments of Zwilling are use in the rejection for Claim 27 and its respective dependant claims. Zwilling (as modified by Hitz) teaches all of Claim 27 with assistance from Zwilling's different embodiments.

**Claim 28** can be mapped to Zwilling (as modified by Hitz) as follows: "The backup system of claim 27, wherein two different nodes of a same storage tree reference a same node in the same storage tree" [Hitz, col. 18, lines 24-30 with Hitz, col. 2, lines 25-36].

**Claim 29** can be mapped to Zwilling (as modified by Hitz) as follows: "The backup system of claim 27, wherein the garbage collection logic is to update references to a node that is within the currently selected range based on an update to an entry in the set of one or more location tables" [Zwilling, Fig. 3C with Zwilling, col. 8, lines 29-33 with Zwilling, col. 9, lines 4-7].

**Claim 30** can be mapped to Zwilling (as modified by Hitz) as follows: "The backup system of claim 27, wherein the garbage collection logic is to prune walking of the nodes of said plurality of storage trees based on the minimum addresses stored in the set of one ore more location tables" [Zwilling, cols. 8-9, lines 46-17 with Zwilling, col. 5, lines 29-52 with Zwilling, col. 12, lines 24-40 with Zwilling, col. 10, lines 5-13].

For **Claim 31**, Zwilling teaches: "An apparatus [Zwilling, cols. 4-5, lines 62-12] comprising:

- ...by recording references to blocks of backed up data [Zwilling, cols. 8-9, lines 46-17 with Zwilling, col. 10, lines 5-13]
- ...an allocator logic to allocate contiguous blocks of storage space from a log of a backup storage space to store said blocks of backed up data [Zwilling, col. 5, lines 29-52 with Zwilling, col. 12, lines 24-40]

- a garbage collection logic...to clean a currently selected contiguous range from the tail of said log, [Zwilling, col. 5, lines 29-52, with Zwilling, Fig. 2] said garbage collection logic to,
  - walk only those nodes of said plurality of storage trees that possibly identify those of said blocks of data that are stored in said currently selected contiguous range or that possibly are themselves stored in said currently selected contiguous range, [Zwilling, cols. 8-9, lines 46-14 with Zwilling, col. 9, lines 43-50]
  - locate blocks of data that are referenced by at least one other block o data residing within the plurality of storage trees, [Zwilling, col. 12, lines 24-53 with Zwilling, Fig. 2 with Zwilling, col. 8, lines 49-51 with Zwilling, col. 8, lines 60-64 with Zwilling, Figs. 3C and 3E] and
  - sweep said currently selected contiguous range, [Zwilling, col. 5, lines 29-52, with Zwilling, Fig. 2] copying blocks of data that are referenced and within the range out of the range and marking the range as unallocated so that at least a portion of the address space within the range can be reclaimed" [Zwilling, col. 5, lines 29-52 with Zwilling, Fig. 2].

Zwilling discloses the above limitations but does not expressly teach:

- "...a hardware backup system to backup a file system, said backup file system including:

- a tracking logic to generate a plurality of storage trees each storage tree having a plurality of nodes and representing backup snapshots of said file system at different times stored in a set of one or more storage devices
- ...responsive to deletion of one or more of said backup snapshots."

With respect to Claim 31, an analogous art, Hitz, teaches:

- "...a hardware backup system [Hitz, col. 17, lines 40-50] to backup a file system, [Hitz, cols. 17-18, lines 65-14] said backup file system including:
  - a tracking logic to generate a plurality of storage trees each storage tree having a plurality of nodes and representing backup snapshots of said file system at different times [Hitz, cols. 17-18, lines 65-14] stored in a set of one or more storage devices [Hitz, cols. 17-18, lines 65-14]
  - ...responsive to deletion of one or more of said backup snapshots" [Hitz, cols. 17-18, lines 65-14].

It would have been obvious to one of ordinary skill in the art at the time of invention having the teachings of Hitz and Zwilling before him/her to combine Hitz with Zwilling because both inventions are directed towards conserving file space (Zwilling shrinks and Hitz only snapshots data that was changed at a block level (not a file level)).

Hitz's invention would have been expected to successfully work well with Zwilling's invention because both inventions use file systems on computers. Zwilling discloses an on-line dynamic file shrink facility comprising shrinking log files. However, Zwilling does not expressly disclose storage trees each representing a snapshot. Hitz

discloses a write anywhere file-system layout comprising storage trees (file system snapshot trees) each representing a snapshot.

It would have been obvious to one of ordinary skill in the art at the time of invention having the teachings of Hitz and Zwilling before him/her to take the write anywhere file-system layout from Hitz and install it into the invention of Zwilling, thereby offering the obvious advantage of taking snapshots of garbage collected/shrunken data, thus saving space. Shrinking the files prior to snapshoting them frees more space in Hitz thus creating more room in Hitz so that another later snapshot won't be prematurely forced out when space in Hitz runs out (by snapshots consuming unacceptable numbers of disk blocks). Also, shinking prior to snapshoting guarantees that Hitz will not back up empty blocks (since they are dirtied by becoming empty in a snapshot). This also conserves space in Hitz. In fact, Hitz desires files to be in a shrunken form (Hitz, col. 5, lines 37-40). Therefore, Hitz is merely using Zwilling's shrinking technique to obtain files that have no fragments.

Different embodiments of Zwilling are use in the rejection for Claim 31 and its respective dependant claims. Zwilling (as modified by Hitz) teaches all of Claim 31 with assistance from Zwilling's different embodiments.

**Claim 32** can be mapped to Zwilling (as modified by Hitz) as follows: "The apparatus of claim 31, wherein the plurality of storage trees include interior nodes and leaf nodes, [Zwilling, col. 8, lines 10-20 with Zwilling, cols. 8-9, lines 46-14] the interior nodes to include references to other nodes in one or more of the plurality of storage trees, [Zwilling, cols. 8-9, lines 46-14] two different interior nodes of a same tree

references a same node in the same tree" [Hitz, col. 18, lines 24-30 with Hitz, col. 2, lines 25-36].

**Claims 33 and 35-39** encompass substantially the same scope of the invention as that of Claims 1, 3-7, respectfully, in addition to a machine-readable medium and some instructions for performing the method steps of Claims 1, 3-7, respectfully. Therefore, Claims 33, 35-39 are rejected for the same reasons as stated above with respect to Claims 1, 3-7, respectfully.

**Claim 50** encompasses substantially the same scope of the invention as that of Claim 18, in addition to a machine-readable medium and some instructions for performing the method steps of Claim 18. Therefore, Claim 50 is rejected for the same reasons as stated above with respect to Claim 18.

14. Claims 8-11, 13-17, 40-43, and 45-49 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,249,792 (Zwilling et al.).

For **Claim 8**, Zwilling teaches: "A method comprising:

- garbage collecting within a range of addresses in a storage system [Zwilling, col. 12, lines 24-53 with Zwilling, Fig. 2] having"

With respect to Claim 8, Zwilling teaches in a different embodiment:

- "a plurality of storage trees, each storage tree having a plurality of nodes and having multiple references to a same block of data, the garbage collecting including: [Zwilling, cols. 8-9, lines 46-17]

- pruning walking of the plurality of storage trees to determine active blocks of data within said range, where active blocks of data are those that are referenced by at least one other block of data still in one of the plurality of storage trees, [Zwilling, cols. 8-9, lines 46-17 with Zwilling, col. 5, lines 29-52 with Zwilling, col. 12, lines 24-53 with Zwilling, Fig. 2 with Zwilling, Figs. 3C and 3E] the pruning walking including:
- determining, based on accessing in one of said plurality of storage trees a parent node that has a plurality of descendent nodes, that none of the plurality of descendant nodes are associated with blocks of data within the range; [Zwilling, col. 5, lines 29-52 with Zwilling, col. 12, lines 24-53 with Zwilling, col. 9, lines 56-65] and
- skipping the walking of the plurality of descendent nodes based on said determining, [Zwilling col. 5, lines 12-16]

- wherein the active blocks determined to be in the range are copied out of the range and the range is marked as unallocated so that at least a portion of the address space within the range can be reclaimed" [Zwilling col. 5, lines 29-52 with Zwilling, Fig. 2].

It would have been obvious to one of ordinary skill in the art at the time of invention to combine the different embodiments of Zwilling because the invention is directed towards shrinking files.

Zwilling discloses an on-line dynamic file shrink facility comprising shrinking log files, however Zwilling does not expressly disclose in that same embodiment how it the

shrinking is accomplished (for instance what type of file a log file is considered as being in Zwilling).

It would have been obvious to one of ordinary skill in the art at the time of invention to take the copying of blocks and file shrinking process from Zwilling and install it into the shrinking log files of Zwilling, thereby offering the obvious advantage of shrinking log files to save space.

Even though Zwilling is only one reference being used to reject Claim 8, the rejection on Claim 8 is under 35 U.S.C. 103(a) because different embodiments of Zwilling are use in the rejection for Claim 8 and its respective dependent claims. Zwilling teaches all of Claim 8 through Zwilling's different embodiments.

**Claim 9** can be mapped to Zwilling as follows: "The method of claim 8, wherein the blocks of data are stored in a log and the range is a segment of the log" [Zwilling, col. 12, lines 15-53 with Zwilling, Fig. 2].

**Claim 10** can be mapped to Zwilling as follows: "The method of claim 9, wherein the segment is at the tail of the log" [Zwilling col. 5, lines 34-52].

**Claim 11** can be mapped to Zwilling as follows: "The method of claim 10, wherein the determining is performed by comparing a minimum offset of the plurality of descendent nodes against the range, [Zwilling, cols. 8-9, lines 46-15 with Zwilling, col. 9, lines 43-51 with Zwilling col. 8, lines 20-35] wherein the minimum offset is accessed when walking the parent node and without walking the plurality of descendent nodes" [Zwilling, cols. 8-9, lines 46-15 with Zwilling, col. 9, lines 43-51 with Zwilling col. 8, lines 20-35 with Zwilling, col. 7, lines 38-21].

**Claim 13** can be mapped to Zwilling as follows: "The method of claim 8, wherein the range is a segment at the tail of a log and said copying is from the said segment at the tail to a segment at the head of the log" [Zwilling col. 5, lines 34-52 with Zwilling, Fig. 2].

**Claim 14** can be mapped to Zwilling as follows: "The method of claim 8, wherein said copying includes updating addresses of the copied blocks of data within a location table" [Zwilling, Fig. 3C with Zwilling, col. 8, lines 29-33 with Zwilling, col. 9, lines 4-7].

For **Claim 15**, Zwilling teaches: "A method of garbage collecting in a storage system [Zwilling, col. 5, lines 29-52 with Zwilling, Fig. 2] comprising."

With respect to Claim 15, Zwilling teaches in a different embodiment:

- "...performing following operations until each block of data that is active in a range to be cleaned at a tail of a log of data is copied to a head of the log, [Zwilling col. 5, lines 34-52 with Zwilling, Fig. 2 with Zwilling, col. 12, lines 15-17] wherein the range to be cleaned is a range of addresses in a storage system having a plurality of storage trees each storage tree having a plurality of nodes, wherein a block of data is associated with a node of the storage tree, [Zwilling, col. 10, lines 5-13 with Zwilling col. 8, lines 46-65] the operations including:
  - locating blocks of data in a log that are both referenced by at least one other block of data residing within the plurality of storage trees, and within the range using pruned walking; [Zwilling, col. 12, lines 24-53 with Zwilling, Fig. 2 with Zwilling, col. 8, lines 49-51 with Zwilling, col. 8, lines 60-64 with Zwilling, Figs. 3C and 3E]

- copying blocks of data associated with child nodes of a current node that are within the range to be cleaned to the head of the log; [Zwilling col. 5, lines 34-52 with Zwilling, Fig. 2]
- retrieving a block of data associated with the current node, upon determining that a minimum address value among addresses of descendent nodes is within the range to be cleaned; [Zwilling col. 5, lines 29-52 with Zwilling, Fig. 2]
- designating, as the current node, one of the child nodes of the current node that is an interior node, upon determining that at least one child node is an interior node; [Zwilling, col. 8, lines 46-65] and
- designating, as the current node, an ancestor node of the current node whose descendent nodes are unprocessed; [Zwilling, col. 8, lines 46-65] and
- marking the range as unallocated when the blocks of data that are active and within the range are copied to the head of the log so that at least a portion of the address space within the range to be cleaned can be reclaimed" [Zwilling col. 5, lines 29-52 with Zwilling, Fig. 2].

It would have been obvious to one of ordinary skill in the art at the time of invention to combine the different embodiments of Zwilling because the invention is directed towards shrinking files.

Zwilling discloses an on-line dynamic file shrink facility comprising shrinking log files, however Zwilling does not expressly disclose in that same embodiment how it the

shrinking is accomplished (for instance what type of file a log file is considered as being in Zwilling)..

It would have been obvious to one of ordinary skill in the art at the time of invention to take the copying of blocks and file shrinking process from Zwilling and install it into the shrinking log files of Zwilling, thereby offering the obvious advantage of shrinking log files to save space.

Even though Zwilling is only one reference being used to reject Claim 15, the rejection on Claim 15 is under 35 U.S.C. 103(a) because different embodiments of Zwilling are use in the rejection for Claim 15 and its respective dependant claims. Zwilling teaches all of Claim 15 through Zwilling's different embodiments.

**Claim 16** can be mapped to Zwilling as follows: "The method of claim 15, wherein performing the following until each block of data that is active in the range to be cleaned at the tail of the log of data is copied to a head of the log includes updating addresses of that copied blocks of data within a location table" [Zwilling, col. 8, lines 21-46 with Zwilling, col. 9, lines 5-7].

**Claim 17** can be mapped to Zwilling as follows: "The method of claim 15, wherein performing the following until each block of data that is active in the range to be cleaned at the tail of the log of data is copied to the head of the log includes updating a minimum address value among addresses of descendent nodes for an entry for the current node in a location table where the minimum address value changes based on copying of the blocks of data associated with the descendent nodes of the current node"

[Zwilling, cols. 8-9, lines 21-15 with Zwilling col. 8, lines 20-35 with Zwilling, col. 7, lines 38-21].

**Claims 40-43, 45, and 46** encompass substantially the same scope of the invention or are rejected for the same rationale as that of Claims 8-11, 13, and 14, respectfully, in addition to a machine-readable medium and some instructions for performing the method steps of Claims 8-11, 13, and 14, respectfully. Therefore, Claims 40-43, 45 and 46 are rejected for the same reasons as stated above with respect to Claims 8-11, 13, and 14, respectfully.

**Claims 47-49** encompass substantially the same scope of the invention as that of Claims 15-17, respectfully, in addition to a machine-readable medium and some instructions for performing the method steps of Claims 15-17, respectfully. Therefore, Claims 47-49 are rejected for the same reasons as stated above with respect to Claims 15-17, respectfully.

15. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

## Conclusion

16. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Brent S. Stace whose telephone number is 571-272-8372 and fax number is 571-273-8372. The examiner can normally be reached on M-F 9am-5:30pm EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Apu M. Mofiz can be reached on 571-272-4080. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Brent Stace B.S.

APU MOFIZ
SUPERVISORY PATENT EXAMINER